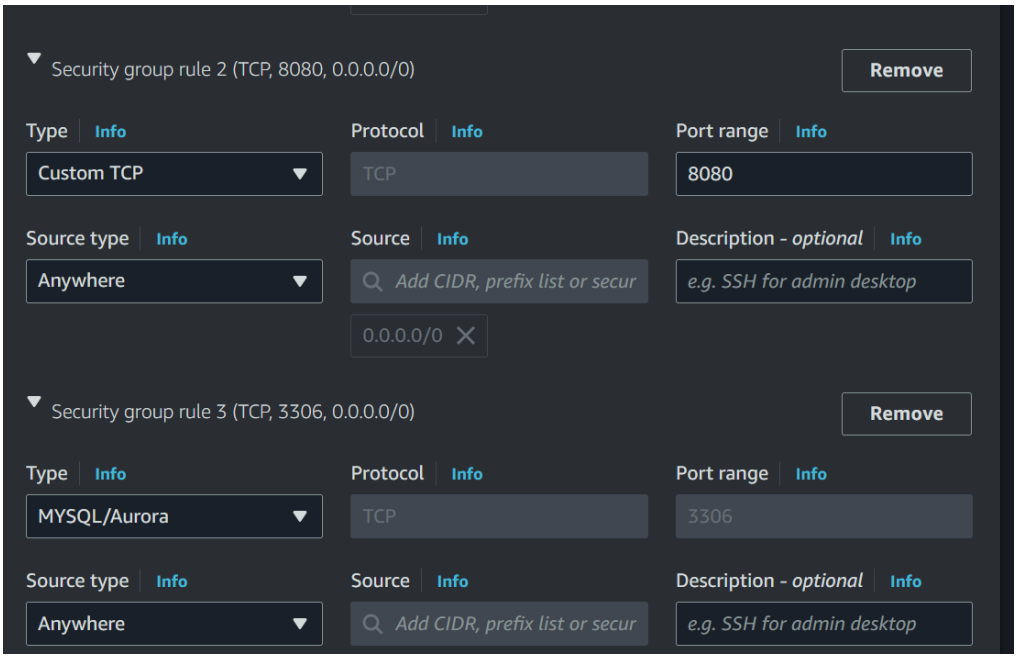
AWS DOCUMENTATION

Host the App

1)Launch Instance with Setting Security Group

Step I : Launch the Instance with adding two rules in Security group .Add Custom TCP : 8080 and MYSQL/Aurora :3306.



Step II : Then Launch Instance and connect it.

2)Set up the Apache Tomcat and Deploy Application.

Step III : Download Apache Tomcat using “curl -O https://dlcdn.apache.org/tomcat/tomcat9/v9.0.87/bin/apache-tomcat-9.0.87.zip” this command.

```
[ec2-user@ip-172-31-25-9 ~]$ curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.87/
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11.7M  100 11.7M    0     0  7292k      0  0:00:01  0:00:01 --:--:-- 7291k
[ec2-user@ip-172-31-25-9 ~]$ ls
apache-tomcat-9.0.87.zip
[ec2-user@ip-172-31-25-9 ~]$ unzip ^C
[ec2-user@ip-172-31-25-9 ~]$ unzip apache-tomcat-9.0.87.zip
```

Step IV : then, unzip it.

Step V : Download Student Application file using “curl -O https://db-tom.s3.ap-northeast1.amazonaws.com/student.war” this command.

Step VI : and then move this file in webapps directory using ““mv student.war apache-tomcat-9.0.87/webapps/” this command.

```
apache-tomcat-9.0.87  apache-tomcat-9.0.87.zip
[ec2-user@ip-172-31-25-9 ~]$ curl -O https://db-tom.s3.ap-northeast-1.amazonaws.com/student.war
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 89423  100 89423    0     0   119k      0 --:--:-- --:--:-- --:--:--  119k
[ec2-user@ip-172-31-25-9 ~]$ ls
apache-tomcat-9.0.87  apache-tomcat-9.0.87.zip  student.war
[ec2-user@ip-172-31-25-9 ~]$ mv student.war apache-tomcat-9.0.87/webapps/
```

Step VII : We need Java for Tomcat to work,so install java using “yum install java -y” command.

```
[ec2-user@ip-172-31-25-9 ~]$ sudo yum install java -y
Last metadata expiration check: 0:20:29 ago on Mon Mar 18 04:51:59 2024.
Dependencies resolved.
================================================================================
 Package                                Architecture        Version
================================================================================
Installing:
```

Step VIII : then,we need to give permission to catalina.sh .

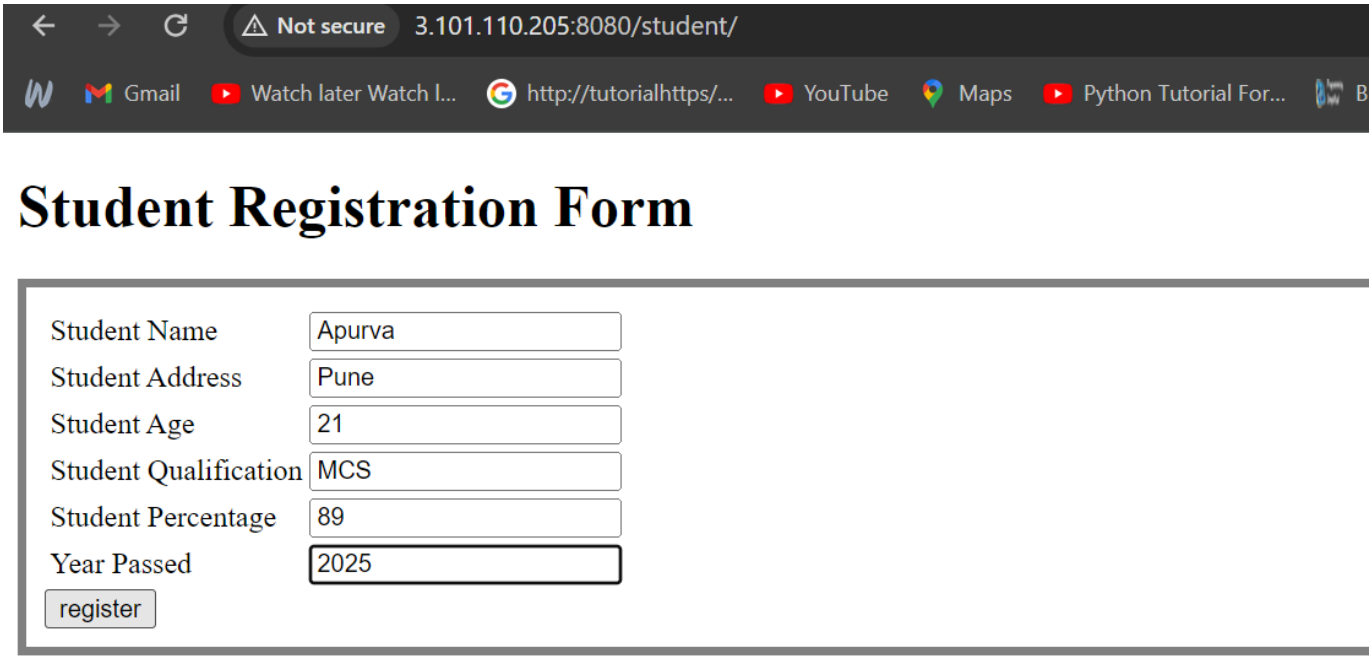
Step IX : for this “ chmod 777 apache-tomcat-9.0.87/bin/catalina.shsudo mv ~/apache-tomcat-9.0.x /usr/lib/tomcat9”using these command to give the execute permissions.

Step X : After giving the permissions ,Start the catalina using “./apache-tomcat-9.0.87/bin/catalina.sh start” these command.

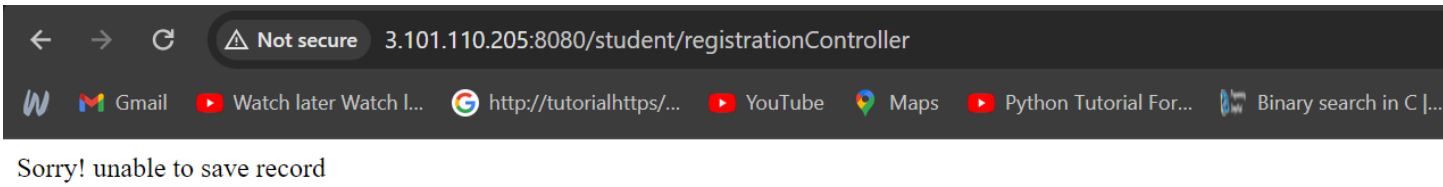
```
Complete!
[ec2-user@ip-172-31-25-9 ~]$ chmod 777 apache-tomcat-9.0.87/bin/catalina.sh
[ec2-user@ip-172-31-25-9 ~]$ ./apache-tomcat-9.0.87/bin/catalina.sh start
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.87/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ec2-user/apache-tomcat-9.0.87/bin/bootstrap.jar:/home/ec2-user/
Using CATALINA_OPTS:
Tomcat started.
```

Step XI : then go to instances, copy the public ip of these Instance and paste it in other browser.

Step XII : and after IP add “:8080/student”.



Step XIII : See here, after registering the data it is unable to save.



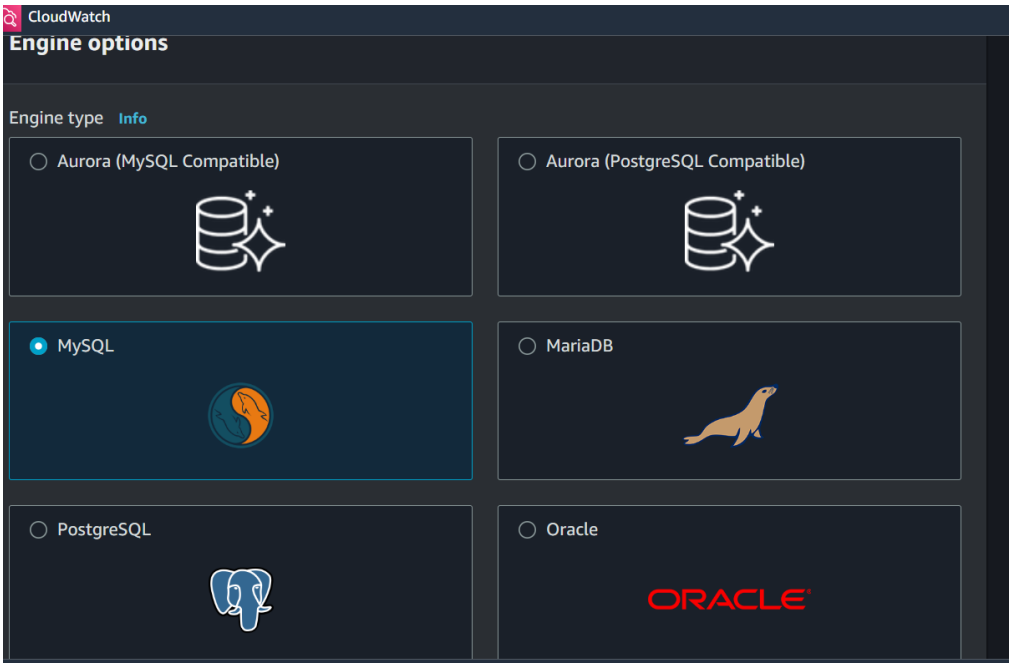
3)Set up the database

Step I : to save the data permanently we need to create database.

Step II : go to the RDS service and click on databases.

Step III : then, create database .

Step IV : Select the Standard create,Engine type is MYSQL, select the free tier template .



Templates

Choose a sample template to meet your use case.

- **Production**
  Use defaults for high availability and fast, consistent performance.
- **Dev/Test**
  This instance is intended for development use outside of a production environment.
- **Free tier**
  Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. **Info**

Settings

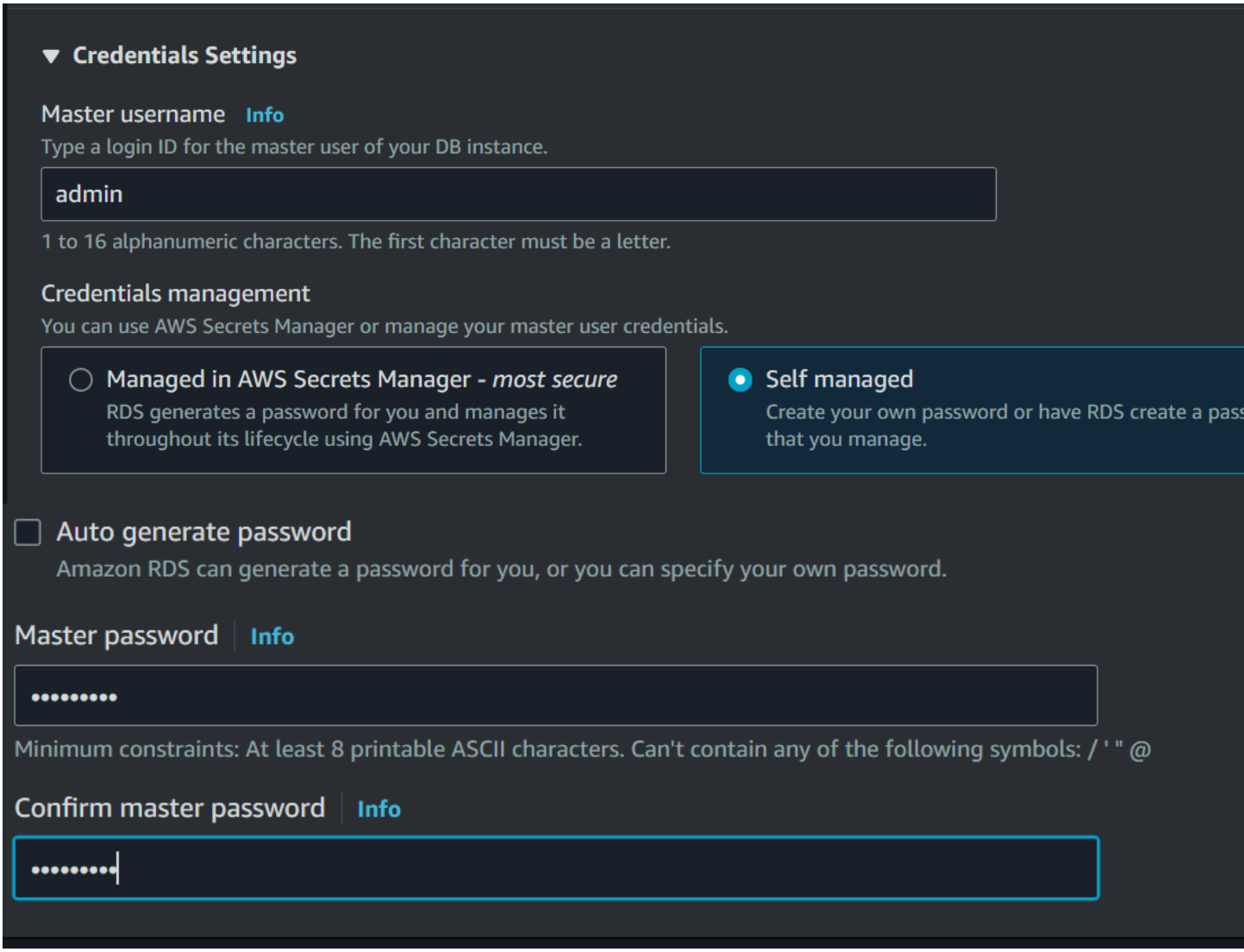
DB instance identifier **Info**

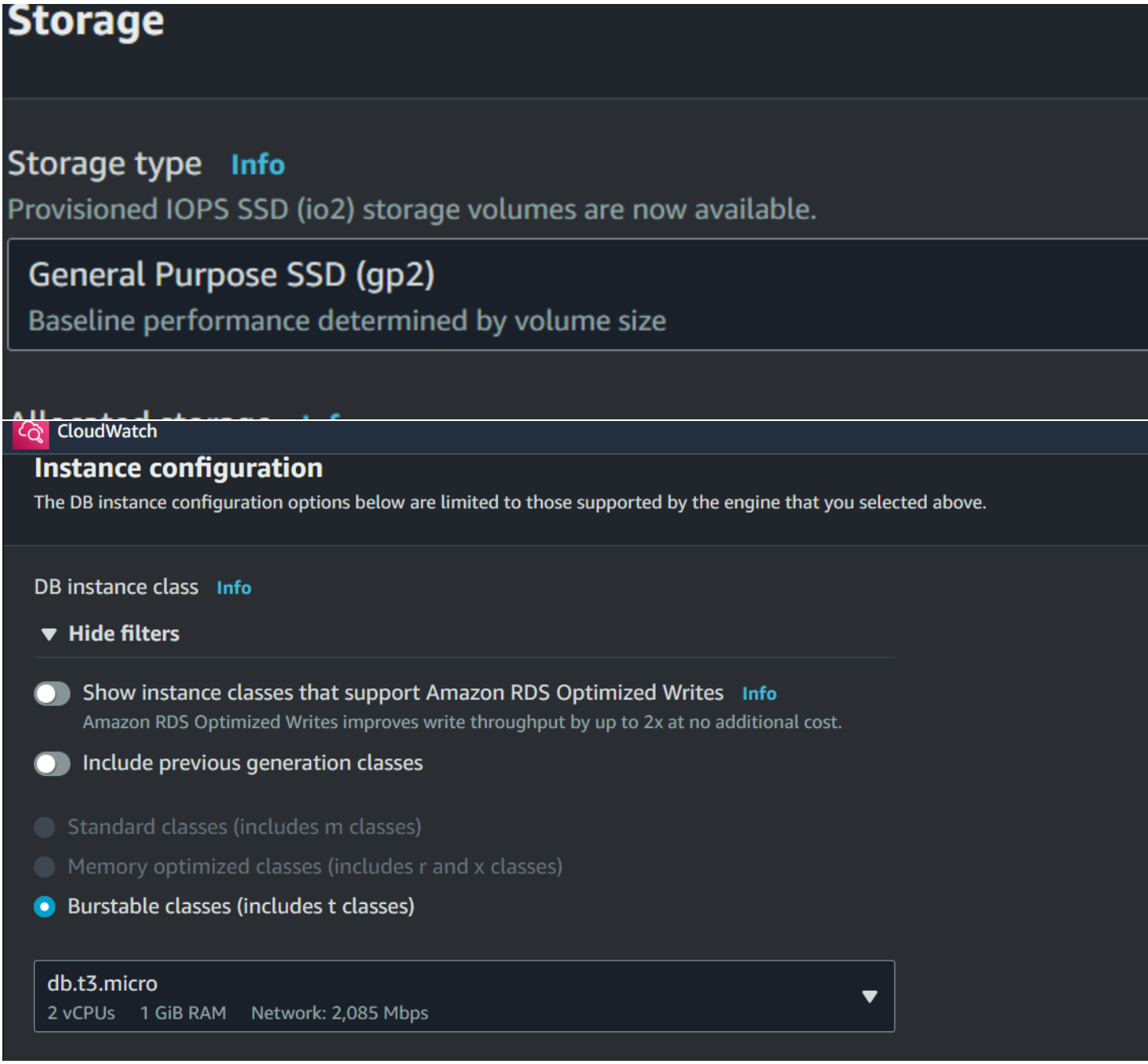
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

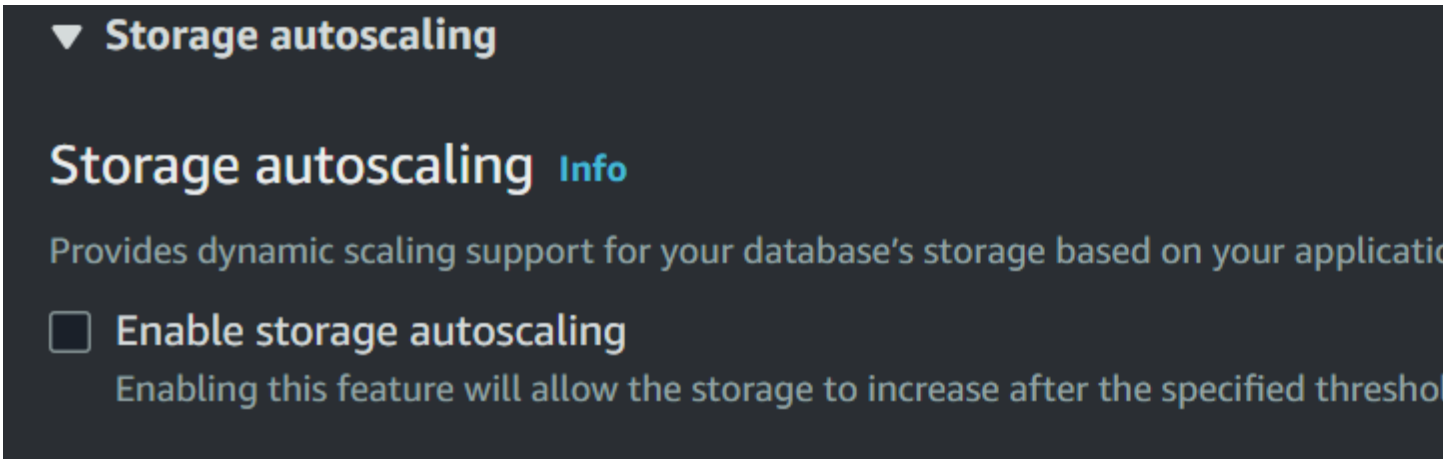
The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Step V : then,enter the name for DB instance, username as a admin and give the Password for database.

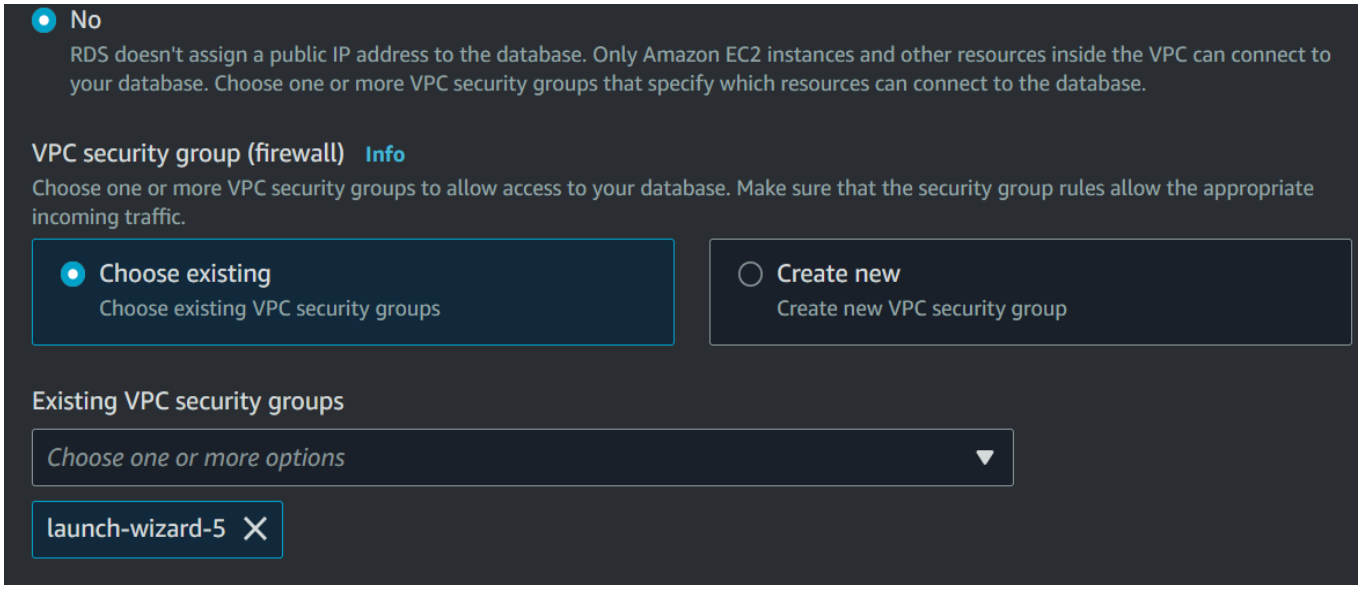




Step VI : then ,uncheck the storage auto scaling option.



Step VII : then choose security group which we used in instance.



Step VIII : then uncheck the Enable automated backups and encryption,auto minor version upgrade and click on create database.

Backup

☐ Enable automated backups
Creates a point-in-time snapshot of your database

Encryption

☐ Enable encryption
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. **Info**

IAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS service-linked role

Maintenance

Auto minor version upgrade **Info**

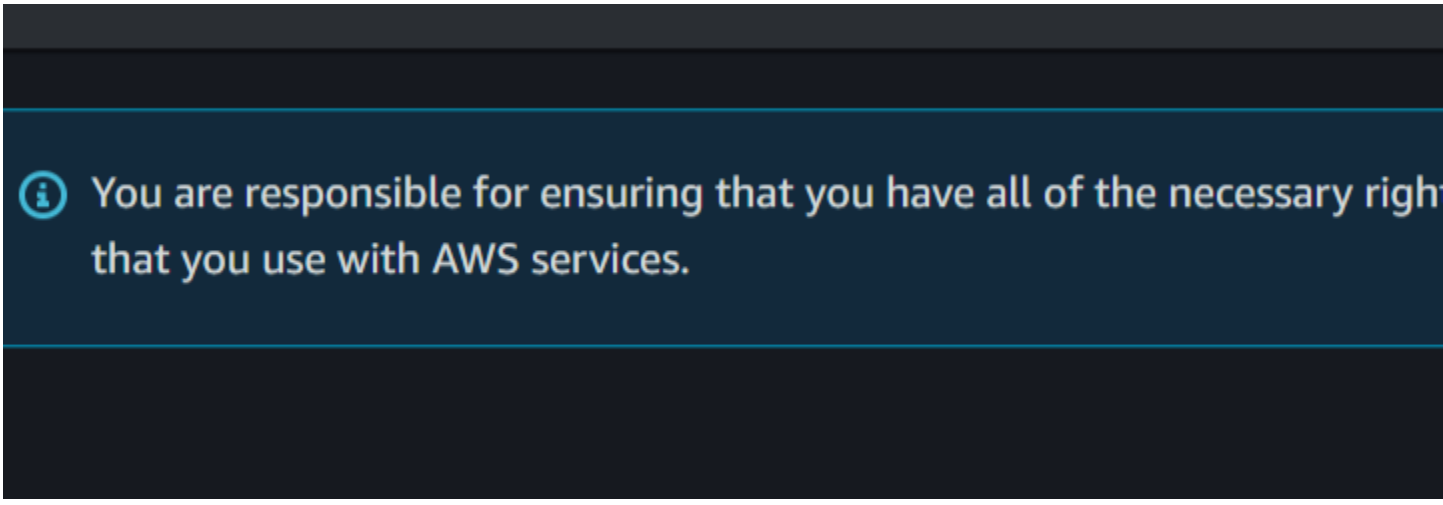
☐ Enable auto minor version upgrade
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window **Info**
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

○ Choose a window
◉ No preference

Deletion protection

☐ Enable deletion protection
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.



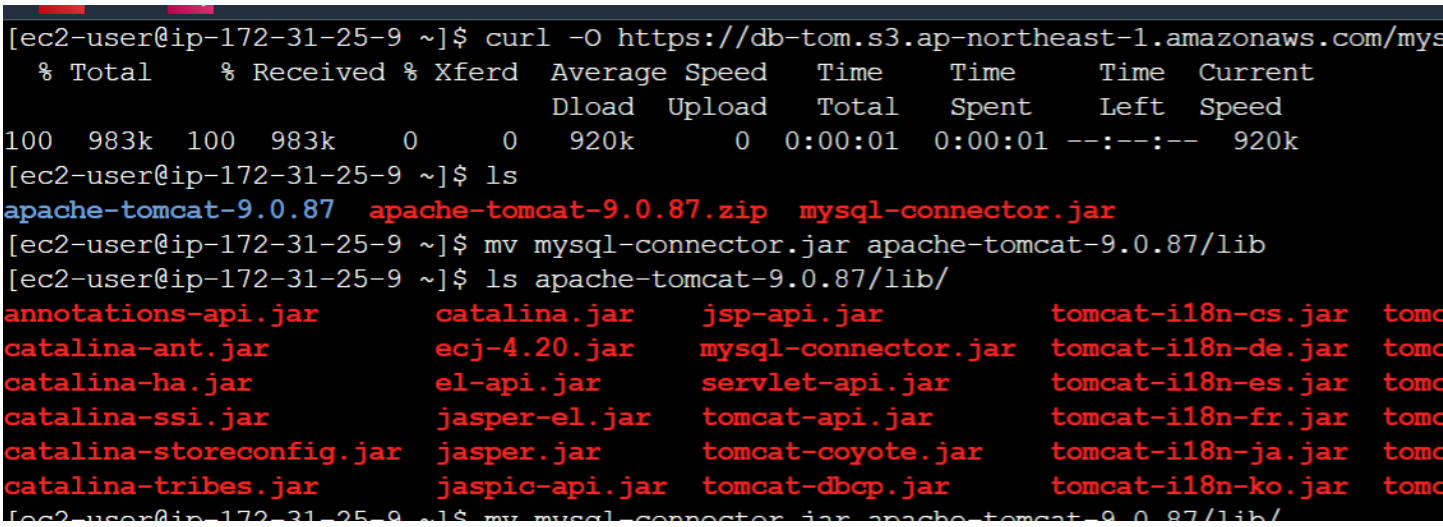
Step IX : see here, database is generated.

4)Configure Tomcat to connect to RDS .

Step I : for connecting we need to MYSQL connector to connect database.

Step II : for downloading MYSQL connector give these command “curl -O https://db-tom.s3.ap-northeast1.amazonaws.com/mysql-connector.jar”.

Step III : Move these file in the lib directory in apache – tomcat, using these “mv mysql-connector.jar apache-tomcat-9.0.87/lib/” command.



Step IV : then edit the tomcat context.xml to add RDS data Configuration in it.

Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource" maxTotal="100" maxIdle="30" maxWaitMillis="10000"

username="admin" password="anvi1234" driverClassName="com.mysql.jdbc.Driver" url="ENDPOINT /DATABASE_NAME"/> • After making the changes in context.xml save the file and exit.

```
<!-- The contents of this file will be loaded for each web application -
<Context>
    <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSo
maxTotal="100" maxIdle="30" maxWaitMillis="10000" username="admin"
password="anvi1234" driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://database-1.cjsuug66g5rg.us-west-1.rds.amazonaws.com/st

    <!-- Default set of monitored resources. If one of these changes, th
    <!-- web application will be reloaded.
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
    <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
    <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
```

Step V : then Install the mariadb using “ yum install mariadb105 -y”.

```
[ec2-user@ip-172-31-25-9 ~]$ sudo vim apache-tomcat-9.0
[ec2-user@ip-172-31-25-9 ~]$ sudo yum install mariadb10
```

5)Connect the Database

Step I : use the Command “mysql -h Endpoint -u username -ppassword;”.

```
IAM  CloudWatch
[ec2-user@ip-172-31-25-9 ~]$ mysql -h database-1.cjsuug66g5rg.us-west-1.rds.amazonaws.com -
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 8.0.35 Source distribution
```

Step II : once connected, then create a new database using “create database database_name;”.

Step III : Switch newly created database using “use db_name”.

Step IV : CREATE TABLE if not exists students(student_id INT NOT NULL AUTO_INCREMENT, student_name VARCHAR(100) NOT NULL, student_addr VARCHAR(100) NOT NULL, student_age VARCHAR(3) NOT NULL, student_qual VARCHAR(20) NOT NULL, student_percent VARCHAR(10) NOT NULL, student_year_passed VARCHAR(10) NOT NULL, PRIMARY KEY (student_id)

);

Add this script in it use exit command to exit form db.

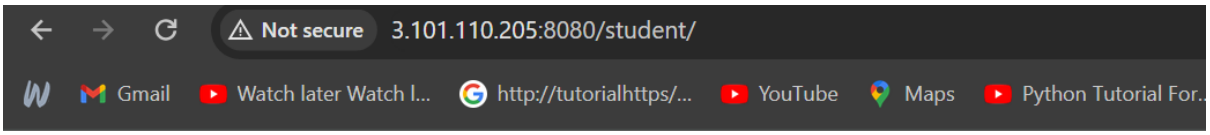
```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> create database studentapp;
Query OK, 1 row affected (0.003 sec)

MySQL [(none)]> use studentapp;
Database changed
MySQL [studentapp]> CREATE TABLE if not exists students(student_id INT NOT NULL
    -> AUTO_INCREMENT,
    -> student_name VARCHAR(100) NOT NULL,
    ->  student_addr VARCHAR(100) NOT NULL,
    -> student_age VARCHAR(3) NOT NULL,
    -> student_qual VARCHAR(20) NOT NULL,
    -> student_percent VARCHAR(10) NOT NULL,
    -> student_year_passed VARCHAR(10) NOT NULL,
    -> PRIMARY KEY (student_id)
    -> );
Query OK, 0 rows affected (0.047 sec)

MySQL [studentapp]> exit
```

Step V : then Stop the catalina.sh using “./apache-tomcat-9.0.87/bin/catalina.sh stop”.

Step VI : add start again using “./apache-tomcat-9.0.87/bin/catalina.sh start”.

```
[ec2-user@ip-172-31-25-9 ~]$ ./apache-tomcat-9.0.87/bin/catalina.sh stop
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.87/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ec2-user/apache-tomcat-9.0.87/bin/bootstrap.jar:/home/ec2-user/ap
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS:  --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=jav
va.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rm
[ec2-user@ip-172-31-25-9 ~]$ ./apache-tomcat-9.0.87/bin/catalina.sh start
Using CATALINA_BASE:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_HOME:   /home/ec2-user/apache-tomcat-9.0.87
Using CATALINA_TMPDIR: /home/ec2-user/apache-tomcat-9.0.87/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ec2-user/apache-tomcat-9.0.87/bin/bootstrap.jar:/home/ec2-user/ap
Using CATALINA_OPTS:
Tomcat started.
[ec2-user@ip-172-31-25-9 ~]$
```

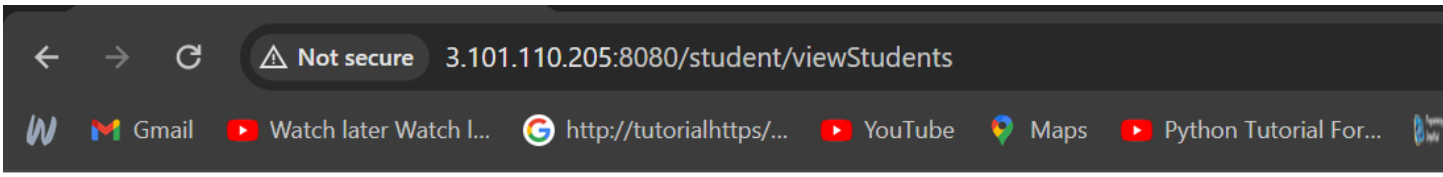
Step VII : see here data will be added successfully.



Student Registration Form

Student Name	anvi
Student Address	pune
Student Age	5
Student Qualification	nur
Student Percentage	99
Year Passed	2024

register



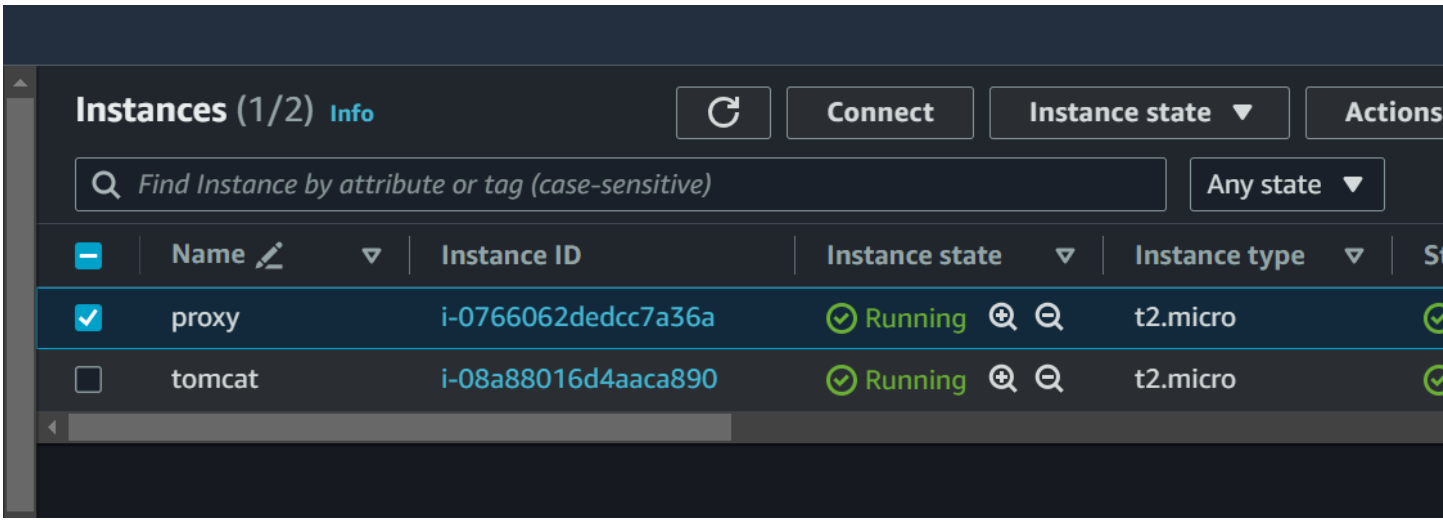
Register Student

Students List

Student ID	StudentName	Student Addrs	Student Age	Student Qualification
1	anvi	pune	5	nur

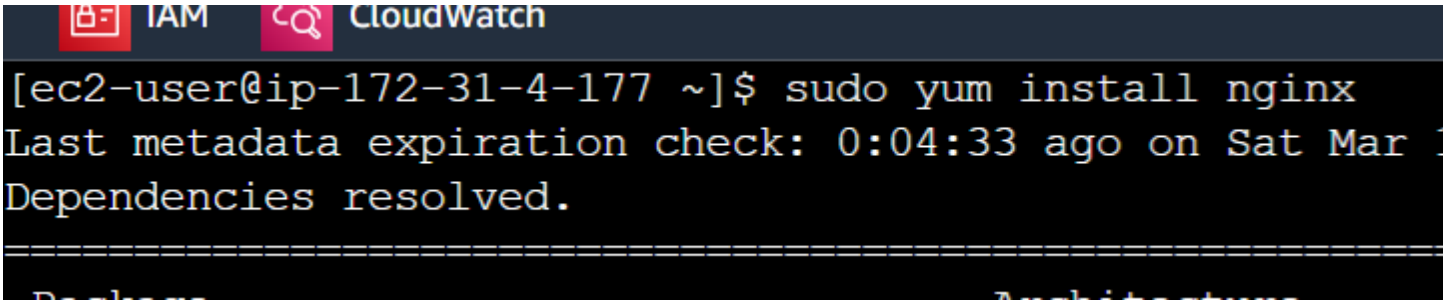
Creating the Proxy server.

Step I : Launch the new Instance add same security group as we added in above instance and also add http.



Step II : Connect the instance .

Step III : then Install nginx.



Step IV : using ls check the file in /etc/nginx.

Step V : then edit the configuration file type “vim /etc/nginx/nginx.conf”.

Step VI : In this, Add the application address in this file.

Location / {

Proxy_pass http: //ApplicationPublicIP:8080/student/;

}

```
location / {
proxy_pass http://3.101.110.205:8080/student/;
}
```

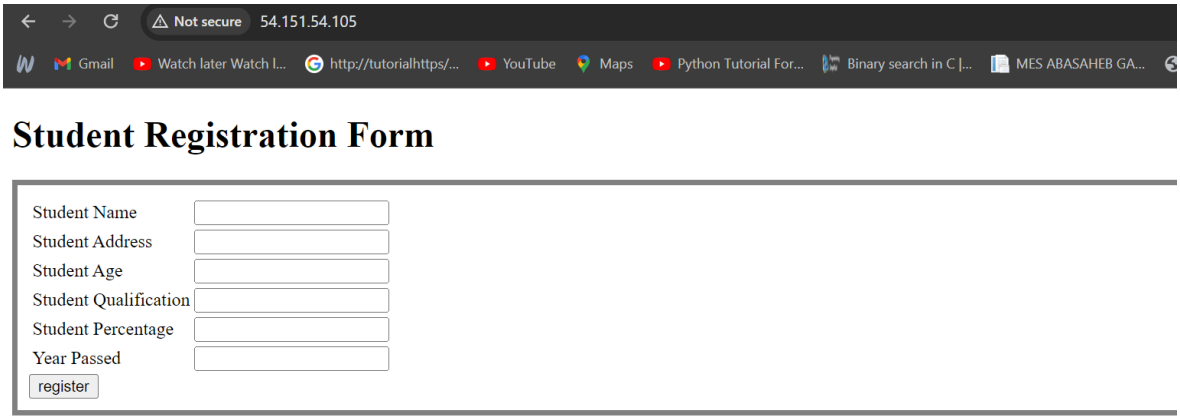
Step VII : then, Start the service using “systemctl start nginx” and enable it.

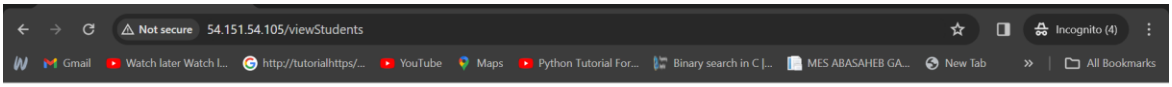
```
[ec2-user@ip-172-31-19-94 ~]$ sudo vim /etc/nginx/nginx.conf
[ec2-user@ip-172-31-19-94 ~]$ sudo su
[root@ip-172-31-19-94 ec2-user]# systemctl start nginx
[root@ip-172-31-19-94 ec2-user]# systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx
[root@ip-172-31-19-94 ec2-user]#
```

To Verify

Copy the public IP of proxy server instance and paste it in another browser.

After registering student see here it is added successfully.





Register Student

Students List

Student ID	StudentName	Student Addrs	Student Age	Student Qualification	Student Percentage	Student Year Passed	Edit	Delete
1	anvi	pune	5	nur	99	2024	edit	delete
2	apurva	mp	20	mcs	89	2025	edit	delete